Computational Statistics
Lab 5

Group 03
Mohsen Pirmoradiyan, Ahmed Alhasan, Yash Pawar

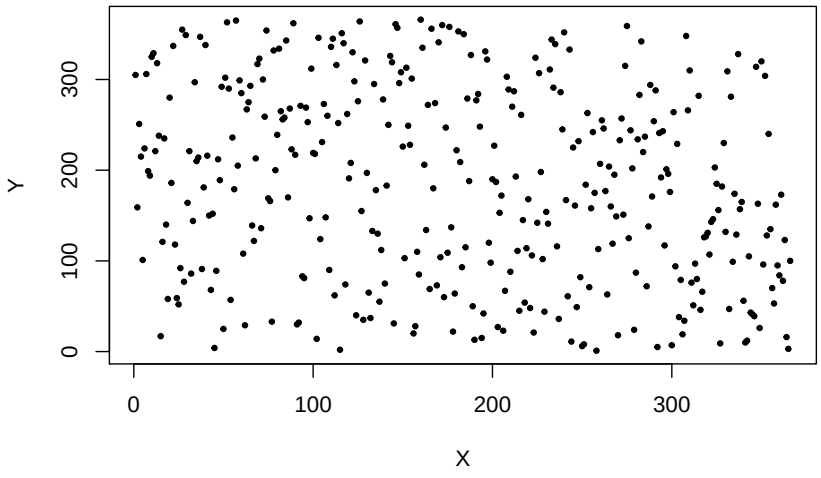
29 Feb 2020

Question 1: Hypothesis testing

In 1970, the US Congress instituted a random selection process for the military draft. All 366 possible birth dates were placed in plastic capsules in a rotating drum and were selected one by one. The first date drawn from the drum received draft number one, the second date drawn received draft number two, etc. Then, eligible men were drafted in the order given by the draft number of their birth date. In a truly random lottery there should be no relationship between the date and the draft number. Your task is to investigate whether or not the draft numbers were randomly selected. The draft numbers (Y=Draft_No) sorted by day of year (X=Day_of_year) are given in the file lottery.xls.

1. Make a scatterplot of Y versus X and conclude whether the lottery looks random.

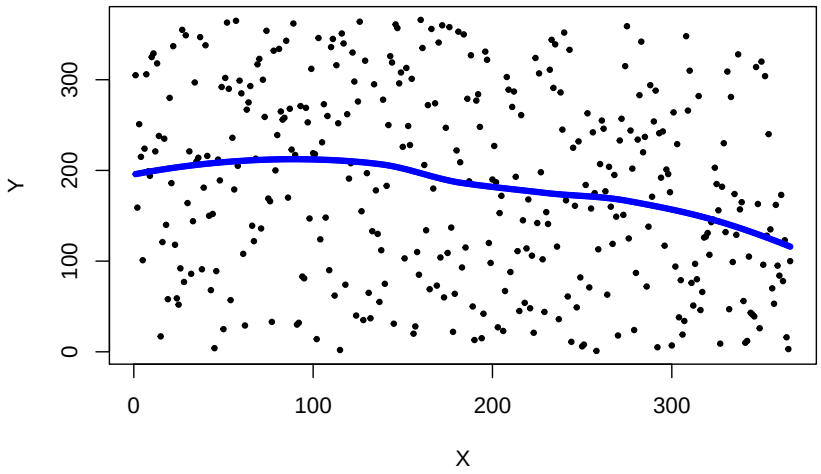
```
RNGversion(min(as.character(getRversion()), "3.6.2"))

library("readxl")
lottery <- read_xls("../Data/lottery.xls")
Y <- lottery$Draft_No
X <- lottery$Day_of_year
plot(X,Y, pch = 19, cex = 0.5)
```



- From the plot it is not possible to tell whether the lottery is random or not.

2. Compute an estimate $\hat{Y}$ of the expected response as a function of X by using a loess smoother (use loess()), put the curve $\hat{Y}$ versus X in the previous graph and state again whether the lottery looks random.

```
plot(X,Y, pch = 19, cex = 0.5)
Y_hat <- loess(Y ~ X)$fitted
points(X, Y_hat, col = "blue", pch = 19, cex = 0.5)
```



- From the loess smoother it sounds like the men with birth dates at the end of the year are less likely to get drafted to military.

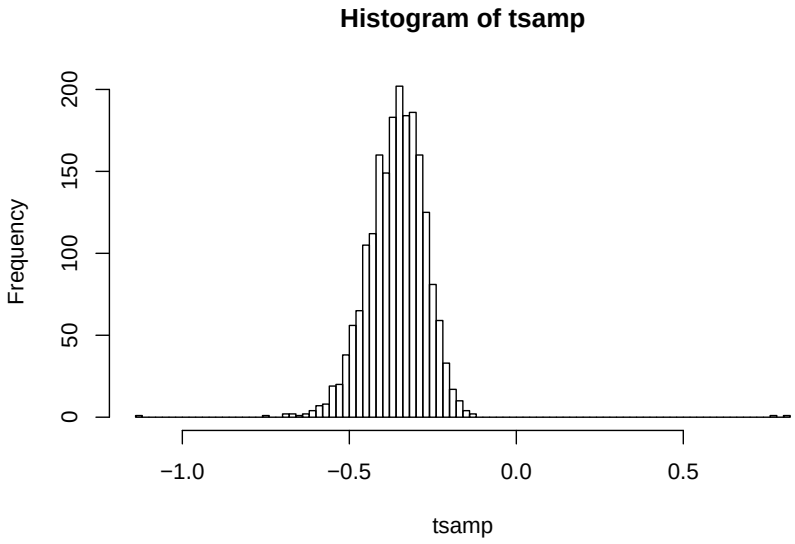
3. To check whether the lottery is random, it is reasonable to use test statistics

$$T = \frac{\hat{Y}(X_b) - \hat{Y}(X_a)}{X_b - X_a}, \quad where\ X_b = argmax_X Y(X),\ X_a = argmin_X Y(X)$$

If this value is significantly greater than zero, then there should be a trend in the data and the lottery is not random. Estimate the distribution of T by using a non-parametric bootstrap with B = 2000 and comment whether the lottery is random or not. What is the p-value of the test?

```
T <- function(X,Y){
  Xb <- X[which.max(Y)]
  Xa <- X[which.min(Y)]
  Y_Xb <- max(Y)
  Y_Xa <- min(Y)

  t <- (Y_Xb - Y_Xa) / (Xb - Xa)
  return(t)
}
```

```r
B <- 2000
n <- dim(lottery)[1]
tsamp <- rep(NA,B)
set.seed(12345, kind="Mersenne-Twister", normal.kind="Inversion")
for (i in 1:B){
  mysample <- lottery[sample(n, size = n, replace = TRUE),]
  X        <- mysample$Day_of_year
  Y        <- loess(mysample$Draft_No ~ mysample$Day_of_year)$fitted
  tsamp[i] <- T(X, Y)
}
hist(tsamp,breaks=100)
```




```r
pvalue <- sum(tsamp > 0) / B
pvalue
```

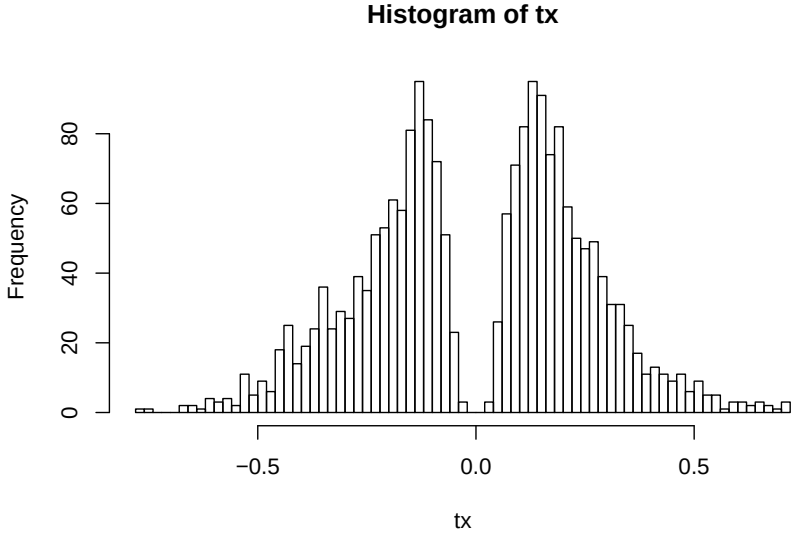
```
## [1] 0.001
```

- With the low p-value we are more likely to reject the null-hypothesis that the lottery is random.

4. Implement a function depending on data and B that tests the hypothesis
   H_0: Lottery is random
   versus
   H_1: Lottery is non-random
   by using a permutation test with statistics T. The function is to return the p-value of this test. Test this function on our data with B = 2000.

```r
test_hypothesis <- function(data, B){
  n  <- dim(data)[1]
  tx <- numeric(B)
  set.seed(12345, kind="Mersenne-Twister", normal.kind="Inversion")
  for (i in 1:B){
```

```
    X     <- sample(n, size = n, replace = TRUE)
    Y     <- loess(data$Draft_No ~ X)$fitted
    tx[i] <- T(X, Y)
  }
  return(tx)
}
tx <- test_hypothesis(data = lottery, B = 2000)
hist(tx,breaks=100)
```




```
Y0 <- loess(lottery$Draft_No ~ lottery$Day_of_year)$fitted
t0 <- T(lottery$Day_of_year, Y0)
pvalue <- length(which(abs(tx) >= abs(t0))) / B
pvalue
```

```
## [1] 0.1515
```

5. Make a crude estimate of the power of the test constructed in Step 4:

(a) Generate (an obviously non-random) dataset with n = 366 observations by using same X as in the original data set and Y (x) = max(0, min(αx + β, 366)), where α = 0,1 and $\beta \sim N(183, sd = 10)$.
(b) Plug these data into the permutation test with B = 200 and note whether it was rejected.
(c) Repeat Steps 5a-5b for α = 0.2,0.3,. . . ,10.

What can you say about the quality of your test statistics considering the value of the power?

```
alpha   <- seq(0.1, 10, 0.1)
pvalues <- vector(length = length(alpha))
for(j in 1:length(alpha)){
  X <- lottery$Day_of_year
  Y <- numeric(366)
  for(i in 1:366){
```

```
    beta <- rnorm(1, mean=183, sd=10)
    YY   <- min((alpha[j] * X[i] + beta), 366)
    Y[i] <- max(0, YY)
  }
  newdata <- data.frame("Day_of_year" = X, "Draft_No" = Y)
  tvec    <- test_hypothesis(data = newdata, B = 200)
  Y0      <- loess(Draft_No ~ Day_of_year, data = newdata)$fitted
  t0      <- T(newdata$Day_of_year, Y0)
  pvalues[j] <- mean(abs(tvec) > abs(t0))
}
length(which(pvalues < 0.05)) / 100
```

```
## [1] 1
```

- The value of the power is 1 meaning all our rejections for the null-hypothesis are correct (no type II error: failure to reject the false null-hypothesis) which solidify the opinion that the lottery is not random.

Question 2: Bootstrap, jackknife and condfidence intervals

The data you are going to continue analyzing is the database of home prices in Albuquerque, 1993. The variables present are Price; SqFt: the area of a house; FEATS: number of features such as dishwasher, refrigerator and so on; Taxes: annual taxes paid for the house. Explore the file prices1.xls.

1. Plot the histogram of Price. Does it remind any conventional distribution? Compute the mean price.

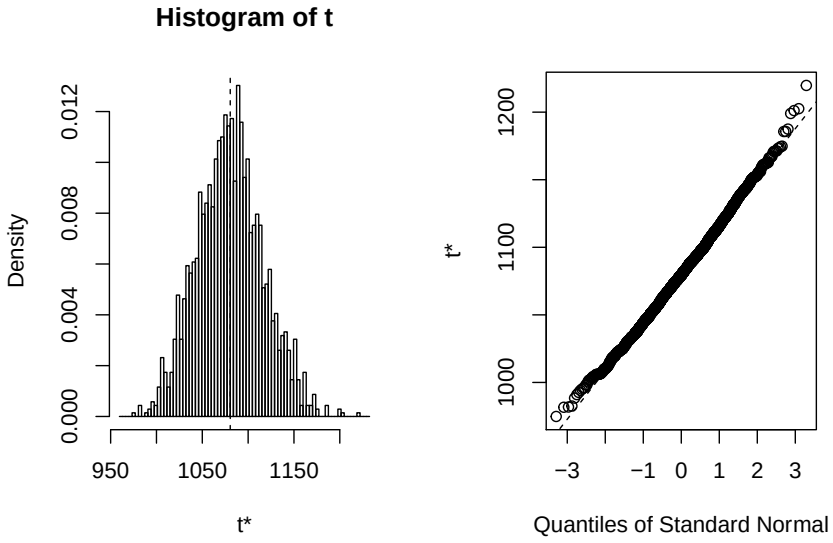
```
prices <- read_xls("../Data/prices1.xls")
hist(prices$Price, breaks = 50)
```


```
mean(prices$Price)
```

```
## [1] 1080.473
```

2. Estimate the distribution of the mean price of the house using bootstrap. Determine the bootstrap bias-correction and the variance of the mean price. Compute a 95% confidence interval for the mean price using bootstrap percentile, bootstrap BCa, and first-order normal approximation (**Hint**: use boot(),boot.ci(),plot.boot(),print.bootci())

```
library(boot)
mymean <- function(data, ind){
  mean(data[ind])
}
set.seed(12345, kind="Mersenne-Twister", normal.kind="Inversion")
samples <- boot(data = prices$Price, statistic = mymean, R = 2000)

plot(samples)
```




```
mean(samples$t)
```

```
## [1] 1080.265
```

```
bias_correction <- 2 * mean(prices$Price) - mean(samples$t)
bias_correction
```

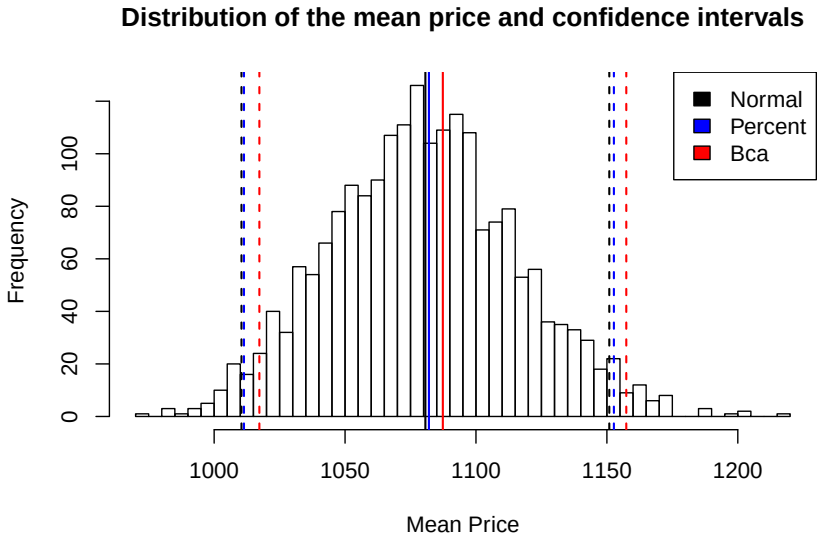
```
## [1] 1080.68
```

```
var(samples$t)[[1]]
```

```
## [1] 1284.346
```

```
conf_ints <- boot.ci(samples)
conf_ints
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = samples)
##
## Intervals :
## Level      Normal              Basic
## 95%   (1010, 1151 )   (1008, 1150 )
##
## Level     Percentile            BCa
## 95%   (1011, 1153 )   (1017, 1157 )
## Calculations and Intervals on Original Scale
```



3. Estimate the variance of the mean price using the jackknife and compare it with the bootstrap estimate

```
Ti <- rep(NA, length(prices$Price))
n  <- length(prices$Price)
for(i in 1:n){
  jack  <- prices$Price[-i]
  Ti[i] <- n * mean(prices$Price) - (n-1) * mean(jack)
}
J_mean <- mean(Ti)
J_mean
```

```
## [1] 1080.473
```

```
J_Var  <- sum((Ti-J_mean)^2) / (n*(n-1))
J_Var
```

```
## [1] 1320.911
```

4. Compare the confidence intervals obtained with respect to their length and the location of the estimated mean in these intervals.

Appendix

```
knitr::opts_chunk$set(echo = TRUE, fig.align = "center", out.width = "70%")
RNGversion(min(as.character(getRversion()), "3.6.2"))

library("readxl")
lottery <- read_xls("../Data/lottery.xls")
Y <- lottery$Draft_No
X <- lottery$Day_of_year
```

```r
plot(X,Y, pch = 19, cex = 0.5)
plot(X,Y, pch = 19, cex = 0.5)
Y_hat <- loess(Y ~ X)$fitted
points(X, Y_hat, col = "blue", pch = 19, cex = 0.5)
T <- function(X,Y){
  Xb <- X[which.max(Y)]
  Xa <- X[which.min(Y)]
  Y_Xb <- max(Y)
  Y_Xa <- min(Y)

  t <- (Y_Xb - Y_Xa) / (Xb - Xa)
  return(t)
}


B <- 2000
n <- dim(lottery)[1]
tsamp <- rep(NA,B)
set.seed(12345, kind="Mersenne-Twister", normal.kind="Inversion")
for (i in 1:B){
  mysample <- lottery[sample(n, size = n, replace = TRUE),]
  X        <- mysample$Day_of_year
  Y        <- loess(mysample$Draft_No ~ mysample$Day_of_year)$fitted
  tsamp[i] <- T(X, Y)
}
hist(tsamp,breaks=100)

pvalue <- sum(tsamp > 0) / B
pvalue
test_hypothesis <- function(data, B){
  n  <- dim(data)[1]
  tx <- numeric(B)
  set.seed(12345, kind="Mersenne-Twister", normal.kind="Inversion")
  for (i in 1:B){
    X     <- sample(n, size = n, replace = TRUE)
    Y     <- loess(data$Draft_No ~ X)$fitted
    tx[i] <- T(X, Y)
  }
  return(tx)
}
tx <- test_hypothesis(data = lottery, B = 2000)
hist(tx,breaks=100)

Y0 <- loess(lottery$Draft_No ~ lottery$Day_of_year)$fitted
t0 <- T(lottery$Day_of_year, Y0)
pvalue <- length(which(abs(tx) >= abs(t0))) / B
pvalue
alpha   <- seq(0.1, 10, 0.1)
pvalues <- vector(length = length(alpha))
for(j in 1:length(alpha)){
  X <- lottery$Day_of_year
  Y <- numeric(366)
  for(i in 1:366){
```

```
    beta <- rnorm(1, mean=183, sd=10)
    YY   <- min((alpha[j] * X[i] + beta), 366)
    Y[i] <- max(0, YY)
  }
  newdata <- data.frame("Day_of_year" = X, "Draft_No" = Y)
  tvec    <- test_hypothesis(data = newdata, B = 200)
  Y0      <- loess(Draft_No ~ Day_of_year, data = newdata)$fitted
  t0      <- T(newdata$Day_of_year, Y0)
  pvalues[j] <- mean(abs(tvec) > abs(t0))
}
length(which(pvalues < 0.05)) / 100
prices <- read_xls("../Data/prices1.xls")
hist(prices$Price, breaks = 50)

mean(prices$Price)
library(boot)
mymean <- function(data, ind){
  mean(data[ind])
}
set.seed(12345, kind="Mersenne-Twister", normal.kind="Inversion")
samples <- boot(data = prices$Price, statistic = mymean, R = 2000)

plot(samples)

mean(samples$t)
bias_correction <- 2 * mean(prices$Price) - mean(samples$t)
bias_correction


var(samples$t)[[1]]

conf_ints <- boot.ci(samples)
conf_ints
hist(samples$t, main="Distribution of the mean price and confidence intervals", xlab="Mean Price", breal
abline(v = conf_ints$normal[2:3],
       col = "black",
       lty = "dashed",
       lwd = 1.5)
abline(v = mean(conf_ints$normal[2:3]),
       col = "black",
       lwd = 1.5)
abline(v = conf_ints$percent[4:5],
       col = "blue",
       lty = "dashed",
       lwd = 1.5)
abline(v = mean(conf_ints$percent[4:5]),
       col = "blue",
       lwd = 1.5)
abline(v = conf_ints$bca[4:5],
       col = "red",
       lty = "dashed",
       lwd = 1.5)
abline(v = mean(conf_ints$bca[4:5]),
```

```
       col = "red",
       lwd = 1.5)
legend(x      = "topright",
       legend = c("Normal", "Percent", "Bca"),
       fill   = c("black", "blue", "red"))
Ti <- rep(NA, length(prices$Price))
n  <- length(prices$Price)
for(i in 1:n){
  jack  <- prices$Price[-i]
  Ti[i] <- n * mean(prices$Price) - (n-1) * mean(jack)
}
J_mean <- mean(Ti)
J_mean
J_Var  <- sum((Ti-J_mean)^2) / (n*(n-1))
J_Var
```